EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	25401719	@ad<"20040112"	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/01 15:19
L2	0	L1 and request$3 with "cache line" with (own$3 ownership) with (processor CPU) with direct$2 same (recall$3 return$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/01 15:19
L3	192	L1 and request$3 with "cache line" with (own$3 ownership) with (processor CPU)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/01 15:19
L4	0	L1 and request$3 same "cache line" with (own$3 ownership) with (processor CPU) same recall$3	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/01 15:19
L5	0	L1 and request$3 with "cache line" with (own$3 ownership) with (processor CPU) same recall$3	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/01 15:22
L6	39	L1 and directory near3 coheren$2 and "cache line" with (own$3 ownership) with direct$2	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/01 15:19
L7	5	L1 and directory near3 coheren$2 and "cache line" with (own$3 ownership) with direct$2 and recall$3	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/01 15:19

EAST Search History

L8	33	L1 and request$3 with "cache line" with (own$3 ownership) with (processor CPU) same (recall$3 transfer$4)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/01 15:19
L9	1163	709/205.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/01 15:19
L10	420	709/212.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/01 15:19
L11	1551	709/232.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/01 15:19
L12	630	712/225.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/01 15:19
L13	2	"6829683".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/01 15:19
L14	59	L1 and request$3 with "cache line" with (own$3 ownership) with (processor CPU) with (recall$3 transfer$4 invalidat$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/01 15:19

EAST Search History

L15	2	"20050033924".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/01 15:19
L16	2	"6381681".pn.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/01 15:19
L17	12	L1 and request$3 with "cache line" with (own$3 ownership) with (processor CPU) with direct$2	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/01 15:19
L18	46	L1 and request$3 with "cache line" with (own$3 ownership) with (processor CPU) same (recall$3 return$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/01 15:19
L19	1218	711/147.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/01 15:19
L20	660	711/169.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/01 15:19
L21	7	L1 and directory near3 coheren$2 and "cache line" with (own$3 ownership) with (processor CPU microprocessor "processing unit") and recall$3	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/01 15:19

EAST Search History

L22	2852	707/200.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/01 15:19
L23	1518	707/201.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/01 15:19
L24	1271	709/213.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/01 15:19
L25	2199	709/230.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/01 15:19
L26	988	(greer.in. schroeder.in. gostin.in.)	US-PGPUB	OR	ON	2007/03/01 16:23
L27	0	26 and (3 6 7 8 14 17 18 21)	US-PGPUB	OR	ON	2007/03/01 16:24
L28	0	1 and 26 and "ownership".clm. and "recall".clm.	US-PGPUB	OR	ON	2007/03/01 16:39
L29	0	1 and 26 and "ownership".clm.	US-PGPUB	OR	ON	2007/03/01 16:41
L30	19	(9 10 11 12 19 20 22 23 24 25) and (3 6 7 8 14 17 18 21)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/03/01 16:43

☐ Search Session History

Thu, 1 Mar 2007, 6:10:25 PM EST

Edit an existing query or compose a new query in the Search Query Display.

Select a search number (#) to:

- Add a query to the Search Query Display
- Combine search queries using AND, OR, or NOT
- Delete a search
- Run a search

Search Query Display

Recent Search Queries

#1 ((processor<in>metadata) <and> (ownership<in>metadata)) <and> (cache line<in>metadata)

#2 ((processor<in>metadata) <and> (ownership<in>metadata)) <and> (recall<in>metadata)

#3 ((cpu<in>metadata) <and> (own<in>metadata))<and> (cache line<in>metadata)

#4 ((recall<in>metadata) <and> (ownership<in>metadata))

#5 ((ownership<in>metadata) <and> (cache line<in>metadata))

#6 ((recall<in>metadata) <and> (return<in>metadata))<and> (request<in>metadata)

#7 ((cache line<in>metadata) <and> (recall<in>metadata))

Indexed by Inspec®

Terms used **ownership cache line processor recall** Found **111** of **198,146**

Sort results by: relevance
Display results: expanded form
Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 111 Result page: **1** 2 3 4 5 6 next

Relevance scale

1 Implementing an untrusted operating system on trusted hardware

David Lie, Chandramohan A. Thekkath, Mark Horowitz
October 2003 **ACM SIGOPS Operating Systems Review , Proceedings of the nineteenth ACM symposium on Operating systems principles SOSP '03**, Volume 37 Issue 5
Publisher: ACM Press

Full text available: pdf(280.87 KB) Additional Information: full citation, abstract, references, citings, index terms

Recently, there has been considerable interest in providing "trusted computing platforms" using hardware~---~TCPA and Palladium being the most publicly visible examples. In this paper we discuss our experience with building such a platform using a traditional time-sharing operating system executing on XOM~---~a processor architecture that provides copy protection and tamper-resistance functions. In XOM, only the processor is trusted; main memory and the operating system are not trusted.Our opera ...

Keywords: XOM, XOMOS, untrusted operating systems

2 An evaluation of memory consistency models for shared-memory systems with ILP processors

Vijay S. Pai, Parthasarathy Ranganathan, Sarita V. Adve, Tracy Harton
September 1996 **ACM SIGPLAN Notices , ACM SIGOPS Operating Systems Review , Proceedings of the seventh international conference on Architectural support for programming languages and operating systems ASPLOS-VII**, Volume 31 , 30 Issue 9 , 5
Publisher: ACM Press

Full text available: pdf(1.64 MB) Additional Information: full citation, abstract, references, citings, index terms

Relaxed consistency models have been shown to significantly outperform sequential consistency for single-issue, statically scheduled processors with blocking reads. However, current microprocessors aggressively exploit instruction-level parallelism (ILP) using methods such as multiple issue, dynamic scheduling, and non-blocking reads. Researchers have conjectured that two techniques, hardware-controlled non-binding prefetching and speculative loads, have the potential to equalize the hardware pe ...

3 The STAMPede approach to thread-level speculation

J. Gregory Steffan, Christopher Colohan, Antonia Zhai, Todd C. Mowry

August 2005 **ACM Transactions on Computer Systems (TOCS)**, Volume 23 Issue 3
Publisher: ACM Press

Full text available: pdf(1.72 MB) Additional Information: full citation, abstract, references, citings, index terms

Multithreaded processor architectures are becoming increasingly commonplace: many current and upcoming designs support chip multiprocessing, simultaneous multithreading, or both. While it is relatively straightforward to use these architectures to improve the throughput of a multithreaded or multiprogrammed workload, the real challenge is how to easily create *parallel software* to allow single programs to effectively exploit all of this raw performance potential. One promising technique fo ...

Keywords: Thread-level speculation, automatic parallelization, cache coherence, chip-multiprocessing

4 A scalable approach to thread-level speculation

J. Greggory Steffan, Christopher B. Colohan, Antonia Zhai, Todd C. Mowry
May 2000 **ACM SIGARCH Computer Architecture News , Proceedings of the 27th annual international symposium on Computer architecture ISCA '00**, Volume 28 Issue 2
Publisher: ACM Press

Full text available: pdf(186.97 KB) Additional Information: full citation, abstract, references, citings, index terms

While architects understand how to build cost-effective parallel machines across a wide spectrum of machine sizes (ranging from within a single chip to large-scale servers), the real challenge is how to easily create parallel software to effectively exploit all of this raw performance potential. One promising technique for overcoming this problem is Thread-Level Speculation (TLS), which enables the compiler to optimistically create parallel threads ...

5 Evaluating the performance of four snooping cache coherency protocols

S. J. Eggers, R. H. Katz
April 1989 **ACM SIGARCH Computer Architecture News , Proceedings of the 16th annual international symposium on Computer architecture ISCA '89**, Volume 17 Issue 3
Publisher: ACM Press

Full text available: pdf(1.70 MB) Additional Information: full citation, abstract, references, citings, index terms

Write-invalidate and write-broadcast coherency protocols have been criticized for being unable to achieve good bus performance across all cache configurations. In particular, write-invalidate performance can suffer as block size increases; and large cache sizes will hurt write-broadcast. Read-broadcast and competitive snooping extensions to the protocols have been proposed to solve each problem. Our results indicate that the benefits of the extensions are limited. Read-broadcast ...

6 Speculative synchronization: applying thread-level speculation to explicitly parallel applications

José F. Martínez, Josep Torrellas
October 2002 **ACM SIGOPS Operating Systems Review , ACM SIGARCH Computer Architecture News , ACM SIGPLAN Notices , Proceedings of the 10th international conference on Architectural support for programming languages and operating systems ASPLOS-X**, Volume 36 , 30 , 37 Issue 5 , 5 , 10
Publisher: ACM Press

Full text available: pdf(1.49 MB) Additional Information: full citation, abstract, references, citings

Barriers, locks, and flags are synchronizing operations widely used programmers and parallelizing compilers to produce race-free parallel programs. Often times, these operations are placed suboptimally, either because of conservative assumptions about the program, or merely for code simplicity.We propose *Speculative Synchronization,* which applies the philosophy behind Thread-Level Speculation (TLS) to explicitly parallel applications. Speculative threads execute past active barriers, busy ...

7 Memory coherence in shared virtual memory systems

Kai Li, Paul Hudak
November 1989 **ACM Transactions on Computer Systems (TOCS)**, Volume 7 Issue 4
Publisher: ACM Press

Full text available: pdf(2.71 MB) Additional Information: full citation, abstract, references, citings, index terms, review

The memory coherence problem in designing and implementing a shared virtual memory on loosely coupled multiprocessors is studied in depth. Two classes of algorithms, centralized and distributed, for solving the problem are presented. A prototype shared virtual memory on an Apollo ring based on these algorithms has been implemented. Both theoretical and practical results show that the memory coherence problem can indeed be solved efficiently on a loosely coupled multiprocessor.

8 Performance analysis of multiprocessor cache consistency protocols using generalized timed Petri nets

Mary K. Vernon, Mark A. Holliday
May 1986 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 1986 ACM SIGMETRICS joint international conference on Computer performance modelling, measurement and evaluation SIGMETRICS '86/PERFORMANCE '86**, Volume 14 Issue 1
Publisher: ACM Press

Full text available: pdf(1.15 MB) Additional Information: full citation, abstract, references, citings, index terms

We use an exact analytical technique, based on Generalized Timed Petri Nets (GTPNs), to study the performance of shared bus cache consistency protocols for multiprocessors. We develop a general framework within which the key characteristics of the Write-Once protocol and four enhancements that have been combined in various ways in the literature can be identified and evaluated. We then quantitatively assess the performance gains for each of the four enhancements. We conside ...

9 Tolerating latency in multiprocessors through compiler-inserted prefetching

Todd C. Mowry
February 1998 **ACM Transactions on Computer Systems (TOCS)**, Volume 16 Issue 1
Publisher: ACM Press

Full text available: pdf(410.70 KB) Additional Information: full citation, abstract, references, citings, index terms, review

The large latency of memory accesses in large-scale shared-memory multiprocessors is a key obstacle to achieving high processor utilization. Software-controlled prefetching is a technique for tolerating memory latency by explicitly executing instructions to move data close to the processor before the data are actually needed. To minimize the burden on the programmer, compiler support is needed to automatically insert prefetch instructions into the code. A key challenge when ...

Keywords: compiler optimization, prefetching

10 Store Memory-Level Parallelism Optimizations for Commercial Applications

Yuan Chou, Lawrence Spracklen, Santosh G. Abraham
November 2005 **Proceedings of the 38th annual IEEE/ACM International Symposium on Microarchitecture MICRO 38**
Publisher: IEEE Computer Society
Full text available: pdf(413.83 KB) Publisher Site Additional Information: full citation, abstract, index terms

This paper studies the impact of off-chip store misses on processor performance for modern commercial applications. The performance impact of off-chip store misses is largely determined by the extent of their overlap with other off-chip cache misses. The epoch MLP model is used to explain and quantify how these overlaps are affected by various store handling optimizations and by the memory consistency model implemented by the processor. The extent of these overlaps are then translated to off-chi ...

11 A single cached copy data coherence scheme for multiprocessor systems
A. Mendelson, D. K. Pradhan, A. D. Singh
December 1989 **ACM SIGARCH Computer Architecture News**, Volume 17 Issue 6
Publisher: ACM Press
Full text available: pdf(667.24 KB) Additional Information: full citation, abstract, index terms

We present and evaluate a snoopy cache memory protocol, the Single Cache Copy Data Coherence (SCCDC), for multiprocessors that allows only a single cache to hold a given share-d data at any time. The simulations presented here indicate that despite its simplicity, the scheme has the potential for good performance comparable with more complex snoopy cache schemes. We have also shown in related work [8] that the existence of only a single copy of data in cache allows efficient access control to sh ...

12 Real-time shading
Marc Olano, Kurt Akeley, John C. Hart, Wolfgang Heidrich, Michael McCool, Jason L. Mitchell, Randi Rost
August 2004 **ACM SIGGRAPH 2004 Course Notes SIGGRAPH '04**
Publisher: ACM Press
Full text available: pdf(7.39 MB) Additional Information: full citation, abstract

Real-time procedural shading was once seen as a distant dream. When the first version of this course was offered four years ago, real-time shading was possible, but only with one-of-a-kind hardware or by combining the effects of tens to hundreds of rendering passes. Today, almost every new computer comes with graphics hardware capable of interactively executing shaders of thousands to tens of thousands of instructions. This course has been redesigned to address today's real-time shading capabili ...

13 Session 6: threads: Thread-Level Speculation on a CMP can be energy efficient
Jose Renau, Karin Strauss, Luis Ceze, Wei Liu, Smruti Sarangi, James Tuck, Josep Torrellas
June 2005 **Proceedings of the 19th annual international conference on Supercomputing ICS '05**
Publisher: ACM Press
Full text available: pdf(370.24 KB) Additional Information: full citation, abstract, references

Chip Multiprocessors (CMP) with Thread-Level Speculation (TLS) have become the subject of intense research. However, TLS is suspected of being too energy inefficient to compete against conventional processors. In this paper, we refute this claim. To do so, we first identify the main sources of dynamic energy consumption in TLS. Then, we present simple energy-saving optimizations that cut the energy cost of TLS by over 60% on average with minimal performance impact. The resulting TLS CMP, populat ...

14 Effects of architectural and technological advances on the HP/Convex Exemplar's

memory and communication performance
Gheith A. Abandah, Edward S. Davidson
April 1998 **ACM SIGARCH Computer Architecture News , Proceedings of the 25th annual international symposium on Computer architecture ISCA '98**, Volume 26 Issue 3
Publisher: IEEE Computer Society, ACM Press
Full text available: pdf(1.42 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

Advances in microarchitecture, packaging, and manufacturing processes enable designers to build new systems with higher performance and scalability. Using microbenchmark techniques, we contrast the memory and communication performance of two generations of the HP/Convex Exemplar scalable parallel processing system. The SPP1000 and SPP2000 have significant architectural and implementation differences, but maintain upward binary compatibility. The SPP2000 employs manufacturing and packaging advanc ...

15 Tradeoffs in buffering speculative memory state for thread-level speculation in multiprocessors
María Jesús Garzarán, Milos Prvulovic, José María Llabería, Víctor Viñals, Lawrence Rauchwerger, Josep Torrellas
September 2005 **ACM Transactions on Architecture and Code Optimization (TACO)**, Volume 2 Issue 3
Publisher: ACM Press
Full text available: pdf(798.26 KB) Additional Information: full citation, abstract, references, index terms

Thread-Level Speculation (TLS) provides architectural support to aggressively run hard-to-analyze code in parallel. As speculative tasks run concurrently, they generate unsafe or speculative *memory state* that needs to be separately buffered and managed in the presence of distributed caches and buffers. Such a state may contain multiple versions of the same variable. In this paper, we introduce a novel taxonomy of approaches to buffer and manage multiversion speculative memory state in mul ...

Keywords: Caching and buffering support, coherence protocol, memory hierarchies, shared-memory multiprocessors, thread-level speculation

16 Cherry-MP: Correctly Integrating Checkpointed Early Resource Recycling in Chip Multiprocessors
Meyrem Kyrman, Nevin Kyrman, Jose F. Martynez
November 2005 **Proceedings of the 38th annual IEEE/ACM International Symposium on Microarchitecture MICRO 38**
Publisher: IEEE Computer Society
Full text available: pdf(453.38 KB) Publisher Site Additional Information: full citation, abstract, index terms

Checkpointed Early Resource Recycling (Cherry) is a recently-proposed micro-architectural technique that aims at improving critical resource utilization by performing aggressive resource recycling decoupled from instruction retirement, using a checkpoint/rollback mechanismto recover from occasional incorrect execution. In this paper, we explore correctness and performance issues that arise when Cherryenabled processors are used in chip multiprocessor architectures. We propose mechanisms to addre ...

17 ReEnact: using thread-level speculation mechanisms to debug data races in multithreaded codes

Milos Prvulovic, Josep Torrellas
May 2003 **ACM SIGARCH Computer Architecture News , Proceedings of the 30th annual international symposium on Computer architecture ISCA '03**, Volume 31 Issue 2
Publisher: ACM Press
Full text available: pdf(184.86 KB) Additional Information: full citation, abstract, references, citings

While removing software bugs consumes vast amounts of human time, hardware support for debugging in modern computers remains rudimentary. Fortunately, we show that mechanisms for Thread-Level Speculation (TLS) can be reused to boost debugging productivity. Most notably, TLS's rollback capabilities can be extended to support rolling back recent buggy execution and repeating it as many times as necessary until the bug is fully characterized. These incremental re-executions are deterministic even i ...

18 A characterization of sharing in parallel programs and its application to coherency protocol evaluation
S. J. Eggers, R. H. Katz
May 1988 **ACM SIGARCH Computer Architecture News , Proceedings of the 15th Annual International Symposium on Computer architecture ISCA '88**, Volume 16 Issue 2
Publisher: IEEE Computer Society Press, ACM Press
Full text available: pdf(1.38 MB) Additional Information: full citation, abstract, references, citings, index terms

In this paper we use trace-driven simulation to analyze the memory reference patterns of write shared data in several parallel applications. We first develop a characterization of write sharing (based on the notion of a write run), and then examine the traces, using metrics derived from the characterization. The results indicate that the amount of write sharing in all programs is small; and that it is characterized by short to medium sequences of per processor references, with little conten ...

19 An adaptive cache coherence protocol optimized for migratory sharing
Per Stenström, Mats Brorsson, Lars Sandberg
May 1993 **ACM SIGARCH Computer Architecture News , Proceedings of the 20th annual international symposium on Computer architecture ISCA '93**, Volume 21 Issue 2
Publisher: ACM Press
Full text available: pdf(1.16 MB) Additional Information: full citation, abstract, references, citings, index terms

Parallel programs that use critical sections and are executed on a shared-memory multiprocessor with a write-invalidate protocol result in invalidation actions that could be eliminated. For this type of sharing, called migratory sharing, each processor typically causes a cache miss followed by an invalidation request which could be merged with the preceding cache-miss request. In this paper we propose an adaptive protocol that invokes this optimization dynamically for migratory b ...

20 Bulk Disambiguation of Speculative Threads in Multiprocessors
Luis Ceze, James Tuck, Josep Torrellas, Calin Cascaval
May 2006 **ACM SIGARCH Computer Architecture News , Proceedings of the 33rd annual international symposium on Computer Architecture ISCA '06**, Volume 34 Issue 2
Publisher: IEEE Computer Society, ACM Press
Full text available: pdf(376.15 KB) Additional Information: full citation, abstract, index terms

Transactional Memory (TM), Thread-Level Speculation (TLS), and Checkpointed multiprocessors are three popular architectural techniques based on the execution of multiple, cooperating speculative threads. In these environments, correctly maintaining data dependences across threads requires mechanisms for disambiguating addresses

across threads, invalidating stale cache state, and making committed state visible. These mechanisms are both conceptually involved and hard to implement. In this paper, ...

Results 1 - 20 of 111 Result page: **1** 2 3 4 5 6 next